

DPS MID EXAM REPORT



Project Title	:	MathUp
Lecturer	:	Ahmad Fadhil

By:

Sarahwati	001202300146
Victoria Astra Nawa	001202300095

**INFORMATICS STUDY PROGRAM
FACULTY OF COMPUTER SCIENCE
PRESIDENT UNIVERSITY**

ABSTRACT

Many students face challenges in identifying their actual math skill level. Traditional testing methods are often perceived as boring, stressful, or inaccessible. This project proposes a web based quiz application that makes math practice more engaging and helps users determine their level in a fun and interactive way.

The application delivers a series of level based math questions, where each level corresponds to a specific school grade ranging from pre kindergarten (pre k) to college, for a total of 15 levels. Users receive one question at a time. If they answer correctly, they gain 2.2 points and move on to the next question or level. A wrong answer ends the game, and the application displays the user's final score and highest level reached. When all 45 questions are completed successfully, users are congratulated for completing the quiz.

This system is built using a React.js frontend for an interactive user interface and a Flask backend for managing logic and data. All components are containerized using Docker, ensuring ease of deployment and scalability. The goal of this project is to provide an accessible, lightweight tool for users, especially students to evaluate their math proficiency in an enjoyable and non-intimidating environment. It can be used for casual practice, informal assessment, or as a supplementary educational resource.

INTRODUCTION

Background & Current Trends

Today, many students find it hard to truly know their actual math skill level. Traditional paper-based tests, which are still widely used in schools, often feel boring, repetitive, or even stressful for some learners. These tests can also be difficult to access outside of a formal classroom setting, making it harder for students to practice math on their own time.

At the same time, there is a growing trend toward using technology to improve education. Many schools and individuals are turning to digital solutions like online quizzes, educational apps, and interactive learning platforms. These tools allow students to study at their own pace, whenever and wherever they want. Instead of feeling pressured by formal exams, learners can now practice through fun, game-like experiences that make learning feel more personal and less intimidating. This shift shows that technology is playing an increasingly important role in helping people improve their skills, especially in subjects like math where regular practice and self-assessment are essential.

Problem Statement

Most students don't have an easy and engaging way to measure their true math skills across different school levels. This makes it hard for them to know which topics they need to practice more.

Objectives

- Help users practice and measure their math skills through level-based questions from pre-kindergarten to college.
- Give 2.2 points for each correct answer and end the game if the answer is wrong.
- Show the final score and highest level reached after the game ends.

Scope & Limitations

- The app includes a limited number of questions, added manually through coding, there is no feature yet for adding new questions through the app interface.
- Multiple choice questions, progress bar, countdown timer, and review of wrong answers are not yet available.
- The app is optimized for desktops and laptops but not fully for mobile devices.

Relevance to Real-World Problems

This project can help anyone measure and improve math skills in a fun way. By making the test interactive and online, it helps reduce barriers to learning and encourages students to keep practicing. MathUp can be used at home or in schools to make math practice more engaging and informative.

METHODS

We built a math quiz application that helps users find out their real math level in a fun way. The quiz uses a level-based system where each level represents a school grade, from pre-kindergarten to college. In total, there are 15 levels with 45 questions.

Our approach was to:

- Show one question at a time to avoid overwhelming the user.
- Let users move to the next question or level only if they answer correctly.
- Give 2.2 points for each correct answer.
- End the game if the answer is wrong, and show their final score and highest level reached.
- If the user answers all questions correctly, show a “Congratulations” screen.

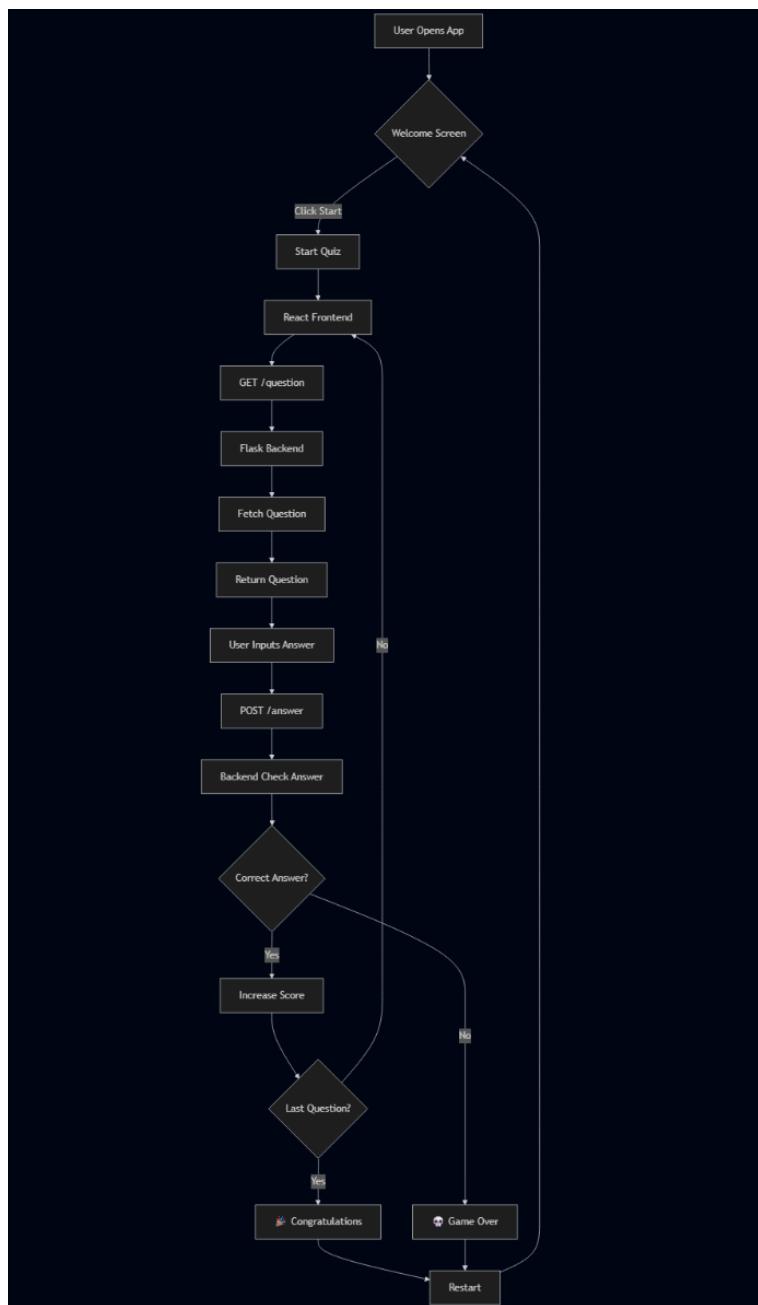
The frontend was created using React.js, where users interact with the quiz, input answers, and view results. The backend was built using Flask (Python), which handles sending the right question based on level and index and checking if the answer is correct or not.

To make setup and sharing easier, we used Docker to package both frontend and backend into containers. This allows the app to run consistently on any system without complex installations. This step-by-step and modular approach keeps the app simple, easy to maintain, and user-friendly.

Development Tools

- Frontend: React.js, JavaScript, HTML, CSS
- Backend: Python, Flask, Flask-CORS
- Containerization: Docker, Docker Compose
- Deployment (local): Docker CLI

Design Diagram



Teamwork & Workflow

Collaboration Platform: GitHub for version control and issue tracking

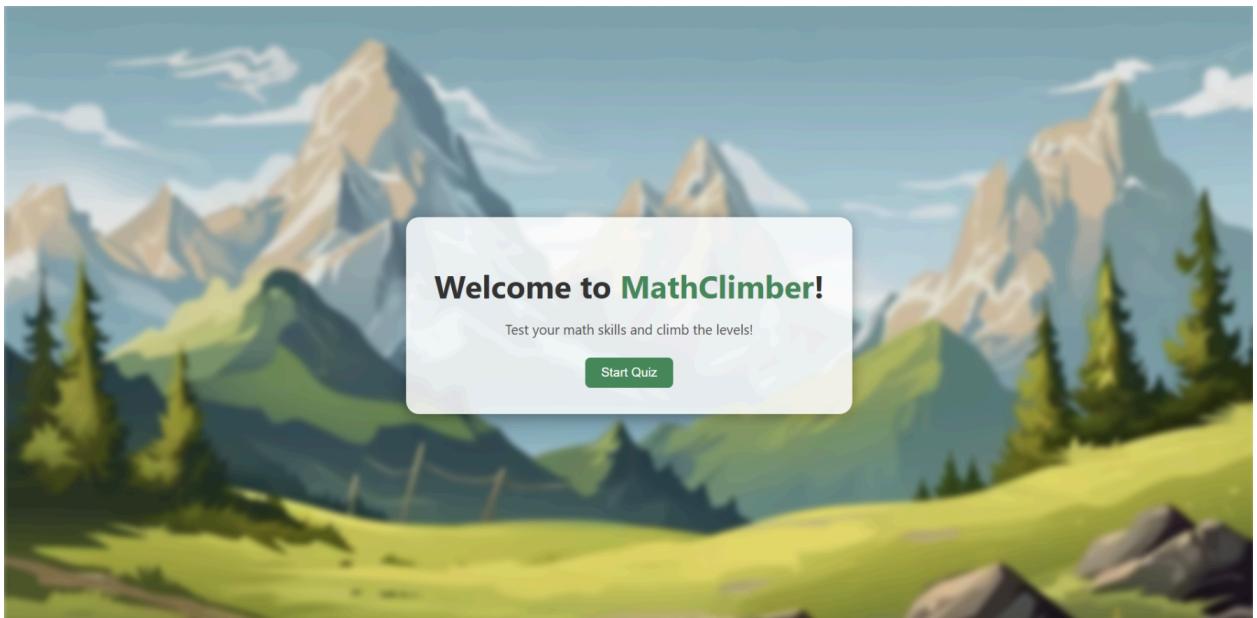
Team Member	Responsibilities	Contributions
Victoria	Frontend Development	<ul style="list-style-type: none">- Designed and implemented the user interface using React JS and CSS- Created start screen, Quiz screen, and game over screen- Connected frontend with backend API to fetch questions and submit answers- Handle input validation, score display, and page transitions- Setup Dockerfile for frontend container- Report writing
Sarah	Backend Development	<ul style="list-style-type: none">- Created the backend using Python, Flask, and Flask-CORS- Developed API endpoints to manage quiz flow- Implement scoring logic and level progression- Manage questions data and added support for image based questions- Setup Dockerfile for the backend container- Report writing

RESULTS

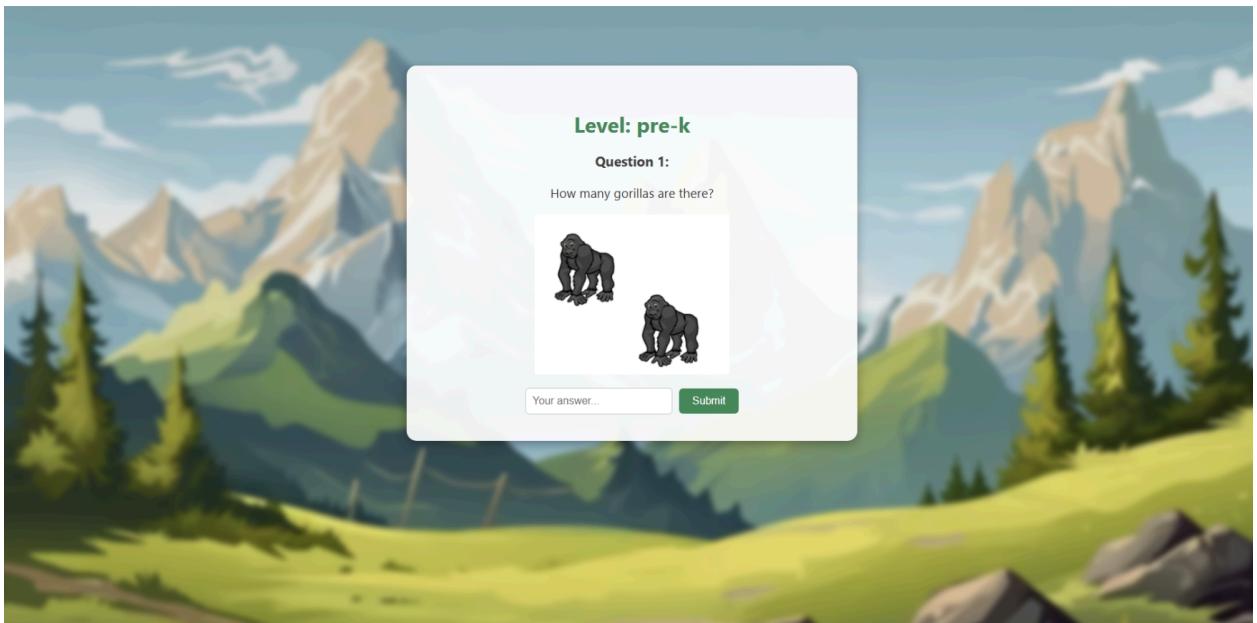
Main Features Developed

- A web based math quiz application with a React frontend and Flask backend.
- Level based question system: users progress through 15 levels, each representing a school grade from pre-kindergarten to college.
- Dynamic question fetching and answer checking with real time feedback.
- Scoring system: users gain 2.2 points for each correct answer.
- Game over feature: the quiz ends immediately upon a wrong answer, and the app shows the final score and highest level reached.
- Containerized deployment with Docker to ensure easy setup and consistent environments.

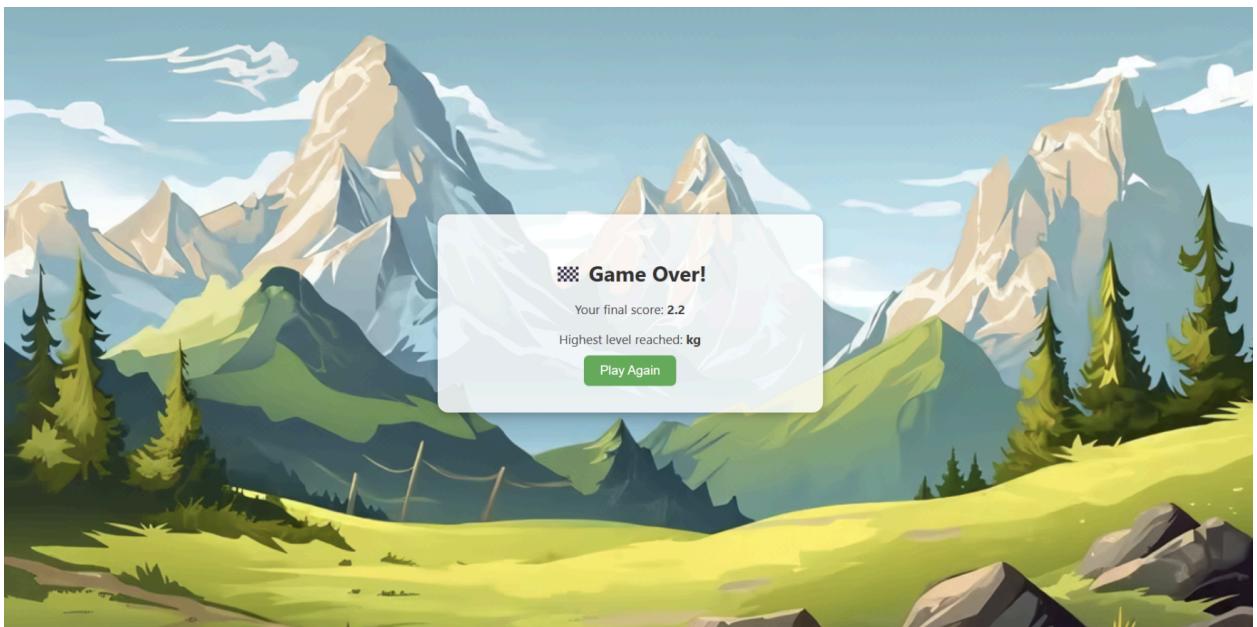
Home Screen / Welcome Screen

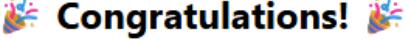


Question Screen



Game Over Screen



 **Congratulations!** 

You completed all questions!

Final Score: **99**

Highest Level: **college**

[Play Again](#)

Risk Assessment Result

- Technical Risks: Low - the app uses well-established technologies (React, Flask).
- Deployment Risks: Low - containerization with Docker reduces environment issues.
- User Risks: Low - no personal data is collected; no privacy concerns.

DISCUSSION

In this project, we successfully created a web based math quiz application called MathUp Climber. The development process gave us several insights into what worked well and what challenges we faced. Overall, the project achieved its main goal, helping users evaluate their math skills in a simple, fun, and non-stressful way.

What Went Well

- The connection between the frontend and backend worked smoothly. Users could answer questions and get real-time feedback without issues.
- Containerizing the app using Docker made the setup easier and helped avoid system compatibility problems.
- The scoring and level progression logic worked as expected, providing users with a clear understanding of their skill level.

What Was Difficult

- One challenge was managing the question data, especially when adding image-based questions. We had to ensure both text and image questions displayed correctly.
- Making sure the frontend and backend communicated properly through API requests also required careful testing, especially when handling wrong answers and showing the game over screen.
- Another difficulty was designing the UI to be clear and responsive. The app is currently optimized for desktops, but not yet for mobile devices.

How Results Compare to Expectations

The app worked as we planned, it shows one question at a time, tracks scores, and ends the game on a wrong answer. Users can easily find their math level in an interactive way. However, some features like multiple choice questions and a progress bar, which we initially hoped to include, are not yet implemented.

Improvements That Could Be Made

- Add multiple choice questions and review features for wrong answers.
- Include a progress bar or countdown timer for more game-like feedback.
- Make the app mobile-friendly so users can play on any device.
- Add a database to store user progress or scores if we want to support account-based features.

Interdisciplinary Challenges

Although this project was mainly focused on web development, managing images and dynamic question data required coordination between frontend and backend. Balancing good design with functional logic was also part of the challenge.

Ethical and Security Considerations

Since the app does not collect personal information, privacy risks are minimal. However, if user accounts or leaderboards were added in the future, we would need to think about protecting user data and ensuring secure authentication.

CONCLUSION

Through this project, we successfully developed MathUp Climber, a web-based quiz application that helps users measure and improve their math skills in a fun and interactive way. The app allows users to progress through level-based questions, score points, and see their final results.

During development, we learned how to connect a React frontend with a Flask backend, manage quiz logic, and package the app using Docker. By using Docker, both the frontend and backend are containerized, making the application easier to set up and deploy on any system without worrying about environment differences. Docker helps ensure the app runs consistently across different devices and is easier to share with others.

For future improvements, we plan to add features such as question management through the app interface, multiple choice options, a progress bar, countdown timer, and full mobile device support. This project shows potential as an accessible and enjoyable learning tool for students of all ages.

Sincerely,

Team

APPENDICES

Full Code Repository:

<https://github.com/Sarahwati-arch/MathUp->

User Manual / Setup Guide:

1. Make sure Docker and Docker Compose are installed.
2. Clone the repository:
`git clone https://github.com/Sarahwati-arch/MathUp-.git`
3. Navigate to the project folder.
4. Run the command:
`docker-compose up --build`
5. Access the app in your browser at <http://localhost:3000>.

Risk Assessment:

- Technical Risks: Low — uses stable technologies (React, Flask, Docker).
- Deployment Risks: Low — Docker minimizes environment compatibility issues
- User Risks: Low — no personal data is collected, so privacy concerns are minimal.